

TITLE OF THE INVENTION

Network With Packet Traffic Scheduling In Response To Quality Of Service and Index
Dispersion Of Counts

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] Not Applicable.

5

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR
DEVELOPMENT

[0002] Not Applicable.

10

BACKGROUND OF THE INVENTION

[0003] The present embodiments relate to computer networks and are more particularly directed to a network with routers or switches configured to schedule traffic according to a dynamic fair mechanism in response to quality of service and an index dispersion of counts.

[0004] As the number of users, traffic volume, and packet speed continue to grow on the global Internet and other networks, an essential need has arisen to provide efficient scheduling mechanisms for packet switched networks. More recently, scheduling has been implicated as needing to consider that the Internet is also evolving towards an advanced architecture that seeks to guarantee the quality of service ("QoS") for real-time applications. QoS dictates the treatment given to packets as they are routed in a network.

One type of QoS framework seeks to provide hard specific network performance guarantees to applications such as bandwidth/delay reservations for an imminent or future data flow. Such QoS is usually characterized in terms of ability to guarantee to an application-specified peak and average bandwidth, delay, jitter, and packet loss. Another 5 type is to use Class-of-Service ("CoS") such as Differentiated Services ("Diff-Serv") to represent the less ambitious approach of giving preferential treatment to certain kinds of packets, but without making any performance guarantees.

[0005] Given the preceding, scheduling mechanisms for packet traffic in switches and routers play a sometimes critical role in providing the QoS guarantees required by many 10 applications such as video-on-demand and multimedia video or teleconferencing. Typical prior art implementations in a router include a number of queues, where packets in each queue belong to a predefined "flow," meaning those packets share one or more predefined attributes. With this structure, classical fair-share scheduling assigns a share of link bandwidth to each queue according to a defined weight for each queue in a fair 15 manner for better QoS implementation. The scheduler chooses in what order service requests can access resources, dictates how to multiplex packets from different connections, and decides which packets to transmit. Various goals are often presented in connection with the scheduling philosophy. For example, a good service discipline should allow the network to treat users differently in accordance with their QoS requirements. As 20 another example, preferably the service discipline can protect packets of well-behaving guaranteed source clients from unconstrained best effort traffic, that is, these sources are given certain bandwidth, yet this flexibility should not compromise the fairness of the scheme to such an extent such that a few classes of users should not be able to degrade service in other classes to the extent that performance guarantees are violated. To allocate 25 bandwidth in a way that QoS of all active flows are satisfied as much as possible, the excess bandwidth of a flow or a class of flows is not only reused by that flow or class, but in some instances it is allocated to other flows or classes as well. The fair-share allocation intends that the flows having little QoS requirements such as Best-Effort traffic can capture the least bandwidth. The fairness for the excess bandwidth allocation can be 30 weighted so that the flows in the higher classes can obtain more bandwidth.

[0006] In general, the prior art scheduling mechanisms fall into two categories, namely, static weight allocation and dynamic weight allocation. Many static schedulers in fast packet routers and switches attempt to provide fair service across a range of traffic classes by employing derivatives of the Generalized Processor Sharing (“GPS”) discipline, in which each of the sessions sharing the link has a first-in first-out (“FIFO”) queue. The scheduler assigns a predetermined weight to each different FIFO queue so that the packets stored in the respective queue are treated according to their assigned weight. However, GPS is limited in that it does not transmit packets as entities, and only one session can receive service at a time and an entire packet must be served before another packet can be served. A typical dynamic scheduling mechanism is the dynamic Weighted Fair Queuing, in which agents in the routers dynamically reconfigure the weights of their associated services. In this scheme, the weights are modified to reflect the changing QoS requirements of a number of packet streams as their queue sizes change over time based on the pre-defined committed information rates.

[0007] While the preceding approaches have merit in some applications, they also include various drawbacks. For example, ideally the traffic scheduler should be influenced by a number of parameters including packet delay and buffer occupancy. However, various static weight allocation mechanisms generally consider little of real-time traffic measurements and QoS information. Instead, they often determine the schedule by sorting the timestamps of packets contending for the link. The exception of brief dynamic behavior in Weighted Fair Queuing itself also focuses around the packet being serviced at that instant in time, and it does not consider the system as a whole and the effect it has on the other sessions later. Further, current dynamic weight allocation mechanisms are not optimized as most of them depend solely on the number of active flows. Although a few of them do consider the QoS information, they merely allocate the excess bandwidth according to the number of flows in a specific class of service or the pre-defined committed information rates.

[0008] In view of the above, there arises a need to address the drawbacks of the prior art, as is accomplished by the preferred embodiments described below.

BRIEF SUMMARY OF THE INVENTION

[0009] In the preferred embodiment, there is a network system, comprising a plurality of nodes. Each node in the plurality of nodes is coupled to communicate with at least one other node in the plurality of nodes. Each node of the plurality of nodes comprises a plurality of queues and is operable to perform the steps of receiving a plurality of packets and, for each received packet in the plurality of packets, coupling the received packet into a selected queue in the plurality of queues, wherein a respective selected queue is selected in response to the respective received packet satisfying one or more criteria. Each node of the plurality of nodes is also operable to perform the step of assigning a weight to each respective queue in the plurality of queues. Each weight assigned to a respective queue in the plurality of queues is responsive to quality requirements for each packet in the respective queue and to a ratio of packet arrival variance in the respective queue and a mean of packets arriving to be stored in the respective queue during a time interval.

[0010] Other aspects are also described and claimed.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

[0011] Figure 1 illustrates a block diagram of a network system 10 into which the preferred embodiments may be implemented.

[0012] Figure 2 illustrates a functional block diagram of preferred aspects of a
5 network packet transfer device of Figure 1.

DETAILED DESCRIPTION OF THE INVENTION

[0013] Figure 1 illustrates a block diagram of a system 10 into which the preferred embodiments may be implemented. By way of introduction, Figure 1 is a functional and logical illustration, that is, it is intended to illustrate the functional operations of a router Rx as well as some of its logical connections, where in certain locations as detailed below actual physical connections are not expressly as shown so as to avoid complicating the data link. Looking then to system 10 in general, it includes a number of stations ST₁ through ST₄, each coupled to a network 20 via a packet transfer device. The term packet transfer device is used in this document in a general sense to refer to any device, typically implemented as a combination of hardware, software, and firmware, that operates to receive a network packet and to place it in one of a number of queues (or buffers), where thereafter the packet transfer device schedules services for the queued packets so as to access resources and so that the packets are taken from the queues and forwarded on to another link within network 20 and ultimately to another station. Such devices are also sometimes referred to as nodes. In an example where network 20 is an internet protocol ("IP") network such as the global Internet or other IP-using network, then each packet transfer device is typically referred to as a router or a switch. However, one skilled in the art should appreciate that the use of the IP protocol is by way of illustration, and many of the various inventive teachings herein may apply to numerous other protocols and packet transfer devices. In any event, returning to network 20 as an IP network, and also by way of an example, each station ST_x may be constructed and function as one of various different types of computing devices, all capable of communicating according to the IP protocol. Lastly and also by way of example, only four stations ST_x are shown so as to simplify the illustration and example, where in reality each such station may be proximate other stations (not shown) and at a geography located at a considerable distance from the other illustrated stations.

[0014] Continuing with Figure 1, and in the example of an IP network, then each packet transfer device along the outer periphery of network 20 is shown as one of edge routers ER₁ through ER₁₁, while within network 20 each packet transfer device is shown as

one of core routers CR₁ through CR₄. The terms edge router and core router are known in the art and generally relate to the function and relative network location of a router. Typically, edge routers connect to remotely located networks and handle considerably less traffic than core routers. In addition and due in part to the relative amount of traffic handled by core routers, they tend to perform less complex operations on data and instead serve primarily a switching function; in other words, because of the tremendous amount of throughput expected of the core routers, they are typically hardware bound as switching machines and not given the capability to provide operations based on the specific data passing through the router. Indeed, core routers typically do not include much in the way of control mechanisms as there could be 10,000 or more connections in a single trunk. In contrast, edge routers are able to monitor various parameters within data packets encountered by the respective router. In any event, the various routers in Figure 1 are shown merely by way of example, where one skilled in the art will recognize that a typical network may include quite a different number of both types of routers. Finally, note that each core router CR_x and each edge router ER_x may be constructed and function according to the art, with the exception that preferably those routers include additional functionality for purposes of traffic routing based on quality of service as considered in packet effective bandwidth, arrival variance, and mean, as described later.

[0015] Completing the discussion of Figure 1, note that the various stations, edge routers, and core routers therein are shown connected to one another in various fashions and also by way of example. Generally characterizing the connections of Figure 1, note that each station ST_x is shown connected to a single edge router ER_x, where that edge router ER_x is connected to one or more core routers CR_x. The core routers CR_x, also by way of example, are shown connected to multiple ones of the other core routers CR_x. By way of reference, the following Table 1 identifies each node (i.e., station or router) shown in Figure 1 as well as the other device(s) to which each is connected.

station or router	connected nodes
ST ₁	ER ₁
ST ₂	ER ₁₀
ST ₃	ER ₅

ST₄	ER₇
ER₁	ST₁; CR₁
ER₂	CR₁; CR₂
ER₃	CR₂
ER₄	CR₂
ER₅	ST₃; CR₂; CR₃
ER₆	CR₃; CR₄
ER₇	ST₄; CR₄
ER₈	CR₄
ER₉	CR₄
ER₁₀	ST₂; CR₁
ER₁₁	CR₁
CR₁	ER₁; ER₂; ER₁₁; ER₁₀; ER₂; CR₂; CR₃; CR₄
CR₂	ER₂; ER₃; ER₄; CR₁; CR₃; CR₄; ER₅
CR₃	ER₅; ER₆; CR₂; CR₁; CR₄
CR₄	ER₇; ER₈; ER₉; CR₁; CR₂; CR₃; ER₆

Table 1

Given the various connections as also set forth in Table 1, in general IP packets flow along the various illustrated paths of network 20, and in groups or in their entirety such packets are often referred to as network traffic. In this regard and as developed below, the 5 preferred embodiments operate such that each router may schedule which packets from the router are transmitted at a given time, in accordance with QoS as well as other considerations.

[0016] Figure 2 illustrates a functional block diagram of certain of the functionality in each router R_x of Figure 1, that is, Figure 2 may be preferably implemented in either or 10 both of edge routers ER_x and core routers CR_x of Figure 1. Note also that the illustration of Figure 2 includes only those blocks deemed helpful in discussing the preferred embodiments, with the further understanding that additional functionality may be applied to any of routers R_x so as to support other known or developed functions provided by a router. Turning then to Figure 2, router R_x includes an input R_{IN} along which packets are

received from network 20, where input R_{IN} thereby represents the physical link connection to the network as well as any associated logical aspects, such as ports or the like. A packet received at input R_{IN} is coupled to an input 30_{IN} of a flow determiner 30. An output 30_{OUT} of flow determiner 30 is connected to provide a received packet to any one of a number 5 $n+1$ packet queues 32_0 through 32_n . In a preferred embodiment, each queue 32_x is a first-in-first-out device and may be constructed according to known principles. The output of each queue 32_x is logically connected to provide each packet to two respective blocks; in practice, the physical connection in this regard may be made by providing a copy of each 10 packet that is input to a queue 32_x also to the two blocks now described, where providing packet copies in this manner allows the true data link through a queue and to the ultimate router output, R_{OUT} , to remain undisturbed so that such traffic may be forwarded directly to a switching matrix (not shown). Looking then to the logical connection of packets from each queue 32_x to two respective blocks, first, the queue output is logically connected to an effective bandwidth ("Eb") estimator 34_x , which estimates a value, Eb , and which as 15 detailed below also produces a corresponding preliminary weight PW_x . Second, the queue output is logically connected to an index dispersion for counts ("IDC") determiner 36_x , which determines a corresponding value IDC_x . The outputs, PW_x and IDC_x , of each pairing of an Eb estimator 34_x and IDC determiner 36_x are connected to a scheduler 38, which represents a logical control function for purposes of scheduling packet service in 20 the various queues 32_0 through 32_n as appreciated in the remainder of this document. Further, and for reasons more clear below, within scheduler 38, the outputs, PW_x and IDC_x , of each pairing of an Eb estimator 34_x and IDC determiner 36_x are connected to a respective multiplier 40_x . The product produced by each of multipliers 40_0 through 40_n is connected to a weight optimizer 42, as is the value of each preliminary weight PW_x . As 25 detailed later, weight optimizer 42 represents a potential adjustment to any of the preliminary weights PW_0 through PW_n to determine final respective weights W_0 through W_n . These final weights are then used to schedule the priority of packet service for the packets in queues 32_0 through 32_n ; in other words, weight W_0 is associated with determining when the packets in queue 32_0 are transmitted, weight W_1 is associated with determining when the packets in queue 32_1 are transmitted, and so forth through W_n being 30

associated with determining when the packets in queue 32_n are transmitted, where each of the transmissions are thus taken from the output of a respective queue 32_x to output Rout of router R_x . Note also that each weight W_x may be said to be associated with a so-called service grant for the respective queue 32_x , where such a grant thereby includes priority, 5 scheduled time, or resources associated with the queue, depending on a specific implementation.

[0017] The operation of router R_x is now described, beginning with flow determiner 30. Flow determiner 30 receives each incoming packet and determines, from a set of criteria, to which one of multiple different flows the packet belongs. Further, for each 10 packet that satisfies a same criterion or criteria, it is routed by flow determiner to a corresponding one of queues 32_0 through 32_n . As a result, each queue 32_x stores packets of a same flow. The criteria evaluated by flow determiner 30 may be based on various different considerations. For example, the criteria may be based on the source and destination address included in the packet. For example with reference to Figure 1, 15 consider the case of core router CR₁ as a router R_x in Figure 2, and consider further that flow determiner 30 of core router CR₁ has three sets of source/destination addresses corresponding to three different respective queues 32_0 , 32_1 , and 32_2 . Also in this example, assume that the first set of source/destination addresses is from station ST₁ to station ST₂, the second set of source/destination addresses is from station ST₁ to station ST₃, and the 20 third set of source/destination addresses is from station ST₁ to station ST₄. Thus, when flow determiner 30 of core router CR₁ receives a packet with a source address of ST₁ and a destination address of ST₂, then flow determiner 30 causes that packet to be stored in queue 32_0 . Also therefore in this example, when flow determiner 30 of core router CR₁ receives a packet with a source address of ST₁ and a destination address of ST₃, then flow 25 determiner 30 causes that packet to be stored in queue 32_1 . Finally, when flow determiner 30 of core router CR₁ receives a packet with a source address of ST₁ and a destination address of ST₄, then flow determiner 30 causes that packet to be stored in queue 32_2 . Note that source and destination addresses are only provided by way of example, where in the preferred embodiment the criteria may be directed to other aspects set forth in the packet 30 header, including by way of example the protocol field, type of service ("TOS") field, or

source/destination port numbers. Moreover, packet attributes about each packet other than that specified in the packet header also may be considered by flow determiner 30. For example, the physical input ports or interfaces connected to other routers may be used by flow determiner 30 as the criteria. In this case and as an instance of this example with

5 reference also to Figure 1, when flow determiner 30 of core router CR₁ receives a packet from edge router ER₁, then flow determiner 30 could cause that packet to be stored in queue 32₀, whereas also in this example, when flow determiner 30 of core router CR₁ receives a packet from edge router ER₂, then flow determiner 30 could cause that packet to be stored in queue 32₁.

10 [0018] As a packet is received in a queue 32_x, certain attributes of the packet are also available to the respective Eb estimator 34_x and IDC determiner 36_x. From these attributes, Eb estimator 34_x estimates the effective bandwidth, *Eb*, for the packet and IDC determiner 36_x determines the value of *IDC* for the packet. The determination of each of these values is discussed below.

15 [0019] Looking now in greater detail to Eb estimator 34_x and its estimation of effective bandwidth, *Eb*, the effective bandwidth for a traffic stream is the minimum bandwidth required for carrying that traffic, subject to meeting QoS requirements. In this regard, and in the context of Figure 2, as a packet arrives its QoS, or the QoS associated with its respective queue 32_x, is available to the respective Eb estimator 34_x. Note that in some 20 instances, the QoS requirements of the traffic are reduced to the condition that a given queue overflow probability not be exceeded. Further, in making these adjustments to QoS, statistical properties of the traffic stream are preferably considered as well as system parameters (e.g., queue size and service discipline) and the traffic mix. Lastly, note that the terms of equivalent bandwidth or equivalent capacity are often used as synonyms for 25 effective bandwidth.

[0020] Given the preceding, a mathematical framework for determining a value of effective bandwidth, *Eb*, has been defined based on the general expression shown in the

following Equation 1, and is noteworthy here insofar as it provides an understanding of the functionality provided by each Eb estimator 32_x in Figure 2:

$$Eb(s, t) = \frac{1}{st} \cdot \log E[e^{(s \cdot A_t)}] \quad \text{Equation 1}$$

In Equation 1, the effective bandwidth is shown as $Eb(s, t)$ to reflect the fact that it relates to variables s and t . In this regard, A_t is the amount of incoming work in duration of t . The values of (s, t) are the so-called space and time parameters, respectively, which characterize the operating point at the router link and depend on the context of the stream (i.e., link resources and the characteristics of the multiplexed traffic). The space parameter s shows the degree of statistical traffic multiplexing or “mix” of the link and the degree of QoS requirements. In this regard, often s tends toward infinity, which corresponds to the case of deterministic multiplexing (i.e., zero probability of overflow), but that case cannot be assumed. If QoS requirements are relaxed, or if the degree of multiplexing increases, s tends to zero and the effective bandwidth, Eb , approaches the mean rate. If QoS requirements are more constrained, or if the degree of multiplexing decreases, s tends to infinity and effective bandwidth, Eb , of the source approaches the maximum rate of $\max(A_t)/t$, measured over the interval t . Note also that the time parameter t corresponds to the most probable duration of buffer busy period prior to overflow.

[0021] The effective bandwidth for various types of traffic models has been derived from the relationship set forth in Equation 1, where examples of such models appear in the following papers, all of which are hereby incorporated herein by reference: (1) C. Courcoubetis and R. Weber, “Buffer overflow asymptotics for a buffer handling many traffic sources,” *J. Appl. Prob.*, vol. 33, pp. 886–903, 1996; (2) G. Kesidis, J. Walrand, and C. S. Chang, “Effective bandwidths for multiclass markov fluids and other ATM sources,” *IEEE Trans. Network*, vol. 1, no. 4, pp. 424–428, Aug. 1993; (3) C. Courcoubetis, V. A. Siris, and G. Stamoulis, “Application of the many sources asymptotic and effective bandwidths to traffic engineering,” *Telecommunication Systems*, vol. 12, no. 2-3, pp. 167–191, 1999; and (4) R. Gibbens and P. Hunt, “Effective bandwidths for the multi-type uas channel,” *Queuing Systems*, vol. 9, pp. 17–28, 1991. However, unlike the estimation of observable

parameters such as mean and variance, the space parameter s parameter cannot be directly estimated from measurements. Accordingly, some effective bandwidths algorithms calculate the space parameter s by using Large Deviations Theory (“LDT”) and by making a large buffer assumption. LDT deals with rare event probabilities and is suitably applied 5 to the effective bandwidth determination since loss probability constraints to be satisfied are very small.

[0022] Note further that other manners exist in the art for estimating effective bandwidth, and those also may be implemented in connection with the preferred embodiment. For example, with space and time parameter estimation being a possible 10 difficulty in the previously mentioned algorithms, *Norros* suggested a different approach to estimate effective bandwidth. This approach does not rely on large deviation theory and it addresses long-range dependent traffic type. This approach is based on the queue analysis of a server with Fractional Brownian Motion (“FBM”) input traffic. The main issue in this method is the FBM parameter estimation. The robust and feasible wavelet 15 based H estimator suits in this method, where “H” is the Hurst parameter, a parameter used to measure the degree of self-similarity behavior on the underlying traffic. In any event, effective bandwidth estimation may be implemented from the above discussion as well as other alternatives and information ascertainable by one skilled in the art.

[0023] As introduced earlier, once an Eb estimator 34_x determines a value for effective 20 bandwidth, Eb_x , then the estimator 34_x also determines a respective preliminary weight, PW_x . More particularly, in the preferred embodiment, for a determined value of effective bandwidth, Eb_x , its respective preliminary weight, PW_x , is as shown in the following Equation 2:

$$PW_x = \frac{Eb_x}{B} \quad \text{Equation 2}$$

25 In Equation 3, B is the total bandwidth available to the router R_x . Thus, the preliminary weight is the ratio of effective bandwidth to total bandwidth. However, in the preferred embodiment and as detailed further below, from each preliminary weight a final and

respective weight, W_x , is determined by weight optimizer 42, and the value of W_x may be adjusted upward above with respect to the respective value PW_x based on two additional consideration, detailed later.

[0024] Looking now in greater detail to IDC determiner 36_x and its determination of a

corresponding value IDC_x , as each packet arrives in a queue 32_x, sufficient packet arrival time corresponding to that packet are stored by the respective IDC determiner 36_x so as to determine the respective value, IDC_x . Particularly, the IDC has heretofore been proposed to be used to characterize packet burstiness in an effort to model Internet traffic, whereas in contrast, in the present inventive scope IDC is instead combined with the other attributes described herein to apply weights to packet queues for purposes of scheduling traffic. By way of background, in the prior art, in a document entitled "Characterizing The Variability of Arrival Processes with Index Of Dispersion," (IEEE, Vol. 9, No. 2, Feb. 1991) by Riccardo Gusella and hereby incorporated herein by reference, there is discussion of using the IDC, which provides a measure of burstiness, so that a model may be described for Internet traffic. Currently in the art, there is much debate about identifying the type of model, whether existing or newly-developed, which will adequately describe Internet traffic. In the referenced document, IDC, as a measure of burstiness, is suggested for use in creating such a model. IDC is defined as the variance of the number of packet arrivals in an interval of length t divided by the mean number of packet arrivals in t . For example, assume that a given network router has an anticipation (i.e., a baseline) of receiving 20 packets per second ("pps"), and assume further that in five consecutive seconds this router receives 30 packets in second 1, 10 packets in second 2, 30 packets in second 3, 15 packets in second 4, and 15 packets in second 5. Thus, over the five seconds, the router receives 100 packets; on average, therefore, the router receives 20 packets per second, that is, the average receipt per second equals the anticipated baseline of 20 pps. However, for each individual second, there is a non-zero variance in the amount of packets received from the anticipated value of 20 pps. For example, in second 1, the variance is +10, in second 2 the variance is -10, and so forth. As such, the IDC provides a measure that reflects this variance, in the form of a ratio compared to its mean, and due to the considerable fluctuation of the receiving rate per second over the five second interval,

there is perceived to be considerable burstiness in the received packets, where the prior art describes an attempt to compile a model of this burstiness so as to model Internet traffic.

[0025] Also in connection with IDC, note that the interval, t , for the present discussion of IDC may be different from the time parameter, t , discussed above for effective bandwidth Eb , and they are not necessarily related. There are various prior art papers that attempt to identify an optimal “ t ” in Eb in the real cases. However, in a preferred embodiment, it may be desirable to align the time interval, t , of IDC with the time parameter, t , of effective bandwidth since scheduling is based on both Eb and IDC, although this is not necessarily the case. For example, the time parameter, t , in Eb can be specified as 2 seconds and the time interval, t , in IDC may be 10 seconds; alternatively, both times can be the same as well if the time scale works in both Eb and IDC.

[0026] Continuing with an examination of IDC determiner 36_x, attention is now directed to its actual operation in determining the IDC value for a given packet. Recalling that the IDC is defined as the variance of the number of packet arrivals in an interval of length t divided by the mean number of packet arrivals in t , it may be written as shown in the following Equation 3:

$$IDC_t = \frac{\text{var}(N_t)}{E(N_t)} \quad \text{Equation 3}$$

In Equation 3, N_t indicates the number of arrivals in an interval of length t . In the preferred embodiment and for estimating the IDC of measured arrival processes, only considered are the time at discrete, equally spaced instants τ_i ($i \geq 0$). Further, letting c_i indicate the number of arrivals in the time interval $\tau_i - \tau_{i-1}$, then the following Equation 4 may be stated:

$$IDC_n = \frac{\text{var}(\sum_{i=1}^n c_i)}{E(\sum_{i=1}^n c_i)} = \frac{\text{var}(c_1 + c_2 + \dots + c_n)}{E(c_1 + c_2 + \dots + c_n)} = \frac{n \cdot \text{var}(c_\tau) + 2 \sum_{j=1}^{n-1} \sum_{k=1}^{n-j} \text{cov}(c_j, c_{j+k})}{n \cdot E(c_\tau)} \quad \text{Equation 4}$$

In Equation 4, $\text{var}(c_\tau)$ and $E(c_\tau)$ are the common variance and mean of c_i , respectively, thereby assuming implicitly that the processes under consideration are at least weakly

stationary, that is, that their first and second moments are time invariant, and that the auto-covariance series depends only on the distance k , the lag, between samples: $\text{cov}(c_i, c_{i+k}) = \text{cov}(c_j, c_{j+k})$, for all i, j , and k .

[0027] Further in view of Equations 3 and 4, consider the following Equation 5:

$$5 \quad \sum_{j=1}^{n-1} \sum_{k=1}^{n-j} \text{cov}(c_j, c_{j+k}) = \text{sum of } \begin{cases} j=1, & \text{cov}(1) + \text{cov}(2) + \dots + \text{cov}(n-2) + \text{cov}(n-1) \\ j=2, & \text{cov}(1) + \text{cov}(2) + \dots + \text{cov}(n-2) \\ \vdots & \vdots \\ j=n-2, & \text{cov}(1) + \text{cov}(2) \\ j=n-1, & \text{cov}(1) \end{cases}$$

$$= \sum_{j=1}^{n-1} (n-j) \text{cov}(j) \quad \text{Equation 5}$$

Further, for the auto-correlation coefficient $\xi_{i,i+k}$, it may be stated as in the following Equation 6:

$$\xi_{i,i+k} = \xi_k = \frac{\text{cov}(c_i, c_{i+k})}{\sqrt{\text{var}(c_i)} \sqrt{\text{var}(c_{i+k})}} = \frac{\text{cov}(k)}{\text{cov}(c_\tau)} \quad \text{Equation 6}$$

10 Then from Equation 6, the following Equation 7 may be written:

$$I_n = \frac{n \cdot \text{var}(c_\tau) + 2 \sum_{j=1}^{n-1} \sum_{k=1}^{n-j} \text{cov}(c_j, c_{j+k})}{n \cdot E(c_\tau)} = \frac{\text{var}(c_\tau)}{E(c_\tau)} \left[1 + 2 \sum_{j=1}^{n-1} \frac{n-j}{n} \xi_j \right]$$

$$= \frac{\text{var}(c_\tau)}{E(c_\tau)} \left[1 + 2 \sum_{j=1}^{n-1} \left(1 - \frac{j}{n} \right) \xi_j \right] \text{by using } \text{cov}(k) = \xi_k \cdot \text{var}(c_\tau) \quad \text{Equation 7}$$

Finally, therefore, the unbiased estimate of $E(c_\tau)$, $\text{var}(c_\tau)$, and ξ_j are as shown in the following respective Equations 8 through 10:

$$E(c_r) = \frac{1}{n} \sum_{i=1}^n c_i \quad \text{Equation 8}$$

$$\text{var}(c_r) = \frac{1}{n-1} \sum_{i=1}^n (c_i - E(c_r))^2 \quad \text{Equation 9}$$

$$\xi_j = \frac{\text{cov}(j)}{\text{var}(c_r)} = \frac{\frac{1}{n-j} \sum_{i=1}^{n-j} (c_i - E(c_r))(c_{i+j} - E(c_r))}{\text{var}(c_r)} \quad \text{Equation 10}$$

Thus, the IDC may be determined by the preferred embodiment using the above
5 Equations 8 and 9, and further in view of Equation 10.

[0028] Having demonstrated various alternatives for preferred embodiment determination of the preliminary weight PW_x and the respective value of IDC_x , recall that each of these values provides a multiplicand into a respective multiplier 40_x , with the product output being connected to weight optimizer 42, which also receives the value of
10 each preliminary weight, PW_x . Given these connections, in the preferred embodiment, weight optimizer 42 is operable to determine, for each preliminary weight, PW_x , a corresponding final weight, W_x . Specifically, these corresponding final weights are determined from the constraints imposed by the following Equations 11, 12, and 13:

$$W_x \geq \left(PW_x = \frac{Eb_x}{B} \right) \quad \text{Equation 11}$$

$$15 \quad \sum_{x=0}^n W_x = 1 \quad \text{Equation 12}$$

$$\min \sum_{x=0}^n W_x * IDC_x \quad \text{Equation 13}$$

Equation 11 demonstrates that for each preliminary weight value, PW_x , its corresponding final weight value, W_x , is equal to or exceeds the preliminary weight value, PW_x . Further, Equation 12 demonstrates that all final weight values combined should total a value of

one. Lastly, Equation 13 solves an objective function, that is, each final weight value, W_x , is adjusted so that the summation of Equation 13 is minimized. This latter constraint therefore is such that by minimizing an objective function from the overall traffic burstiness, each value W_x is determined so as to fairly allocate the bandwidth weight to
5 smooth the bursty traffic without compromising the QoS requirements. Given the final weight values $\{W_0, W_1, \dots, W_n\}$, weight optimizer 42 outputs those as part of scheduler 38, to control respective queues 32_0 through 32_n . In other words, each queue 32_x is then serviced in priority as defined by its corresponding final weight, W_x . Accordingly,
10 scheduling of resource access and packet transmission is thereby weighted according to these values and, thus, this more fairly allocates bandwidth while smoothing burstiness and taking into consideration QoS requirements

[0029] From the above illustrations and description, one skilled in the art should appreciate that the preferred embodiments provide a computer network with routers or switches configured to schedule traffic according to a dynamic fair mechanism in response
15 to quality of service and an index dispersion of counts. The embodiments provide numerous benefits over the prior art. As one example, as compared to static mechanisms, the preferred embodiments dynamically schedule link bandwidth based on real-time traffic measurements. In addition, unlike various dynamic algorithms, which simply allocate excess bandwidth according to the number of flows in a specific class of service or
20 the pre-defined committed information rates, the preferred embodiment considers the actual on-line traffic burstiness, as measured in IDC, as an objective function. As still another example, the preferred embodiments also take advantage of effective bandwidth as the lower bound, which guarantees the QoS requirements for the high priority traffic flows or classes can be always satisfied during the optimization. As yet another benefit, in
25 the preferred embodiments, excess bandwidth of a flow or a class of flows is not only reused by that flow or class but is allocated to other flows or classes as well. Further, by designating the lower bound for the flows having little QoS requirements such as Best-Effort traffic, these flows can also capture the least bandwidth so that the fairness for the excess bandwidth allocation can be achieved. Note that these preferred embodiments and
30 benefits can be well applied in the DiffServ environment because in that context the

classes of traffic flows are the primary targets instead of individual flows so that there are less scalability issues. As a final benefit, while the preferred embodiments have been described in connection with an IP network, they also may be applied to any network that is cell or packet based. Given the above, it will be further appreciated that while the 5 present embodiments have been described in detail, various substitutions, modifications or alterations could be made to the descriptions set forth above without departing from the inventive scope which is defined by the following claims.